**1. Безопасный удаленный доступ**

Иногда мы получаем доступ к нашему серверу удаленными методами и если это никак  не защищено, мы можем подвергнуть наш сервер опасности.

Опциями OpenSSH управляют через конфигурационный файл /etc/ssh/sshd\_config.

**a. Отключите метод аутентификации по паролю SSH и включите метод аутентификации с открытым ключом**

Измените значения ChallengeResponseAuthentication и PasswordAuthentication на ‘no’, чтобы деактивировать метод пароля.

PasswordAuthentication no

ChallengeResponseAuthentication no

Авторизация по открытому ключу:

RSAAuthentication yes

PubkeyAuthentication yes

Теперь вы можете сгенерировать новую пару ключей:

# ssh-keygen -t rsa

**b. Отключите прямой вход root’a в систему**

Риск безопасности существует, если  позволен  вход пользователя root непосредственно к серверу.

Вместо этого вы должны войти в систему под вашей учетной записью и затем сделать su – чтобы войти в систему как root.

Таким образом, вы должны изменить PermitRootLogin ‘yes’ на PermitRootLogin ‘no’.

PermitRootLogin no

**c. Измените порт по умолчанию для SSH (например: 8500)**

port 8500

**2. Пароль на загрузчик GRUB**

Защищая загрузчик мы можем предотвратить доступ к однопользовательскому режиму, в котором пользователь входит в систему автоматически как root.

Для Debian:

# grub-mkpasswd-pbkdf2

Для Centos:

# grub2-mkpasswd-pbkdf2

**3. Порты сети**

После конфигурирования сетевых служб важно обратить внимание, которое порты на самом деле прослушиваются на сетевых интерфейсах системы. Любые открытые порты могут быть доказательством проникновения.

# nmap -sT -O localhost

Starting Nmap 6.40 ( http://nmap.org ) at 2017-06-07 23:13 UTC

Nmap scan report for localhost (127.0.0.1)

Host is up (0.000061s latency).

Other addresses for localhost (not scanned): 127.0.0.1

rDNS record for 127.0.0.1: centos-01

Not shown: 995 closed ports

PORT STATE SERVICE

22/tcp open ssh

25/tcp open smtp

80/tcp open http

111/tcp open rpcbind

2049/tcp open nfs

Чтобы перечислить все открытые порты и связанные программы используйте команду ниже:

# netstat -tulpn

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 0 0.0.0.0:111 0.0.0.0:\* LISTEN 1/systemd

tcp 0 0 0.0.0.0:20048 0.0.0.0:\* LISTEN

**4. Снижение прав**

Сузьте права для системных файлов и папок, чтобы ограничить риски.

# chmod 700 /root

# chmod 700 /var/log/audit

# chmod 740 /etc/rc.d/init.d/iptables

# chmod 740 /sbin/iptables

# chmod -R 700 /etc/skel

# chmod 600 /etc/rsyslog.conf

# chmod 640 /etc/security/access.conf

# chmod 600 /etc/sysctl.conf

**5. Проверьте аккаунты на пустые пароли**

Любая учетная запись, имеющая пустой пароль, означает, что она открыта для несанкционированного доступа в сети, подрывая безопасность сервера Linux.

Чтобы проверить наличие пустых паролей используйте команду ниже:

# cat /etc/shadow | awk -F: '($2==""){print $1}'

paul

Заблокируйте пустые пароли:

# passwd -l paul

Locking password for user paul.

passwd: Success

**6. Настройте параметры ядра**

Отредактируйте файл /etc/sysctl.conf, чтобы оптимизировать параметры ядра.

Sysctl – команда, используемая, чтобы изменить параметры ядра во время выполнения в терминале.

# sysctl -a

# sysctl -A

# sysctl net.ipv4.conf.all.rp\_filter

Чтобы подгрузить настройки введите:

# sysctl -p

Скопируйте следующий контент в файл /etc/sysctl.conf:

# Turn on execshield

kernel.exec-shield=1

kernel.randomize\_va\_space=1

# Enable IP spoofing protection

net.ipv4.conf.all.rp\_filter=1

# Disable IP source routing

net.ipv4.conf.all.accept\_source\_route=0

# Ignoring broadcasts request

net.ipv4.icmp\_echo\_ignore\_broadcasts=1

net.ipv4.icmp\_ignore\_bogus\_error\_messages=1

# Make sure spoofed packets get logged

net.ipv4.conf.all.log\_martians = 1

net.ipv4.conf.default.log\_martians = 1

# Disable ICMP routing redirects

sysctl -w net.ipv4.conf.all.accept\_redirects=0

sysctl -w net.ipv6.conf.all.accept\_redirects=0

sysctl -w net.ipv4.conf.all.send\_redirects=0

sysctl -w net.ipv6.conf.all.send\_redirects=0

# Disables the magic-sysrq key

kernel.sysrq = 0

# Turn off the tcp\_sack

net.ipv4.tcp\_sack = 0

# Turn off the tcp\_timestamps

net.ipv4.tcp\_timestamps = 0

# Enable TCP SYN Cookie Protection

net.ipv4.tcp\_syncookies = 1

# Enable bad error message Protection

net.ipv4.icmp\_ignore\_bogus\_error\_responses = 1

**7. Отключите нежелательные службы**

Вам необходимо удалить все нежелательные службы и демоны (службы, выполняющиеся в фоновом режиме) из системы, которые запускаются во время загрузки на уровне запуска 3.

# chkconfig --list | grep '3:on'

Чтобы отключить службы, выполните:

# service serviceName stop

# chkconfig serviceName off

**8. Требовать аутентификацию для однопользовательского режима**

При входе в однопользовательский режим рекомендуется вводить пароль root.

Откройте файл / etc / sysconfig / init и добавьте строку:

SINGLE=/sbin/sulogin

**9. Улучшенная безопасность Linux (SELinux)**

SELinux – это набор правил безопасности, которые определяют, какой процесс может получить доступ к файлу, каталогам, портам и т. Д.

Каждый файл, процесс, каталог и порт имеют специальную метку безопасности, называемую контекстами SELinux.

Контекст – это просто имя, которое используется политикой SELinux, чтобы определить, может ли процесс получить доступ к файлу, каталогу или порту.

По умолчанию политика не позволяет взаимодействовать, поэтому явные правила предоставляют доступ.

Если правила разрешений отсутствуют, доступ запрещен.

Команда getenforce сообщает нам, в каком режиме находится SELinux.

Мы можем изменить режим SELinux в принудительном порядке, изменив SELINUX=enforcing в /etc/sysconfig/selinux

В этом файле есть три директивы, как описано ниже.

* **Enforcing**: политика безопасности SELinux применяется.
* **Permissive**: SELinux печатает предупреждения
* **Disabled**: SELinux полностью отключен.

Вы можете проверить статус SELinux командой:

# sestatus

SELinux status: disabled

Сейчас мы видим статус disabled, чтобы включить применение политик безопасности, выполним команду:

# setenforce enforcing

**10. Установите брандмауэр с помощью iptables**

iptables – это прикладная программа, которая позволяет системному администратору настраивать таблицы, предоставленные брандмауэром ядра Linux, а также цепочки и правила, которые он хранит.

**a. Закройте все нежелательные порты**

# iptables -A INPUT -p tcp --dport PORT\_NUMBER -j DROP

**b. Заблокируйте плохие IP**

#  iptables -A INPUT -s IP\_ADDRESS -j DROP

**с. Блокировать подключения к сетевому интерфейсу**

Чтобы заблокировать подключения с определенного IP-адреса к определенному сетевому интерфейсу, используйте команду:

# iptables -A INPUT -i <span class="highlight">ens0</span> -s 6<span class="highlight">.6.6.6</span> -j DROP

**d. Список правил iptables**

Вы можете увидеть все правила iptables командой

# iptables -L -n -v

**11. Проверка файловой системы**

Все файлы с битами SUID / SGID могут использоваться для вредоносных действий, когда исполняемый файл SUID / SGID имеет проблему безопасности.

Любой локальный или удаленный пользователь может использовать такой файл.

**а. Определить нежелательные бинарники SUID и SGID**

# find / \( -perm -4000 -o -perm -2000 \) -print

# find / -path -prune -o -type f -perm +6000 -ls

**b. Определение файлов, доступных для записи**

# find /dir -xdev -type d \( -perm -0002 -a ! -perm -1000 \) -print

**c. Определение сиротских файлов и папок**

# find /dir -xdev \( -nouser -o -nogroup \) -print

**12. Держать каталог /boot как только для чтения**

Ядро Linux и связанные с ним файлы находятся в каталоге / boot, который по умолчанию используется как и на чтение, так и на запись.

Изменение его на чтение снижает риск несанкционированной модификации важных загрузочных файлов.

Нам нужно отредактировать файл / etc / fstab и вставить строку ниже <

LABEL=/boot /boot ext2 defaults,ro 1 2

**13. Отключить все TCP оболочки**

Оболочки TCP могут обеспечить быстрый и простой способ контроля доступа к связанным с ними приложениям.

Поэтому рекомендуется блокировать все неиспользуемые приложения, а затем разрешать только те приложения, которые будут использоваться.

Например, мы заблокируем все приложения, но санкционируем ssh

# echo"ALL:ALL" >> /etc/hosts.deny

# echo "sshd:ALL" >> /etc/hosts.allow

**14. Заблокируйте cronjobs для неавторизованных пользователей.**

Cron используется для автоматизации заданий в определенное время.

Можно указать, кто может и не сможет выполнять задания.

Это контролируется с помощью файлов под названием /etc/cron.allow и /etc/cron.deny.

Чтобы заблокировать пользователя с помощью cron, просто добавьте имена пользователей в cron.deny и чтобы разрешите пользователю запускать cron add в файле cron.allow.

# echo ALL >> / etc / cron.deny

**15. Защитите сервер от переполнения буфера**

Переполнение буфера происходит тогда, когда программа или процесс пытается записать большее количество данных в блок фиксированной длины памяти или буфер, чем сам буфер, выделенный для хранения.

Важно защитить ваш сервер от этой атаки.

**а. Включите ExecShield**

# sysctl -w kernel.exec-shield=1

Вы также можете добавить строку ниже в /etc/sysctl.conf:

kernel.exec-shield = 1

**b. Проверить / включить ASLR**

Ранжирование пространства адреса – это защитная функция, которая затрудняет переполнение буфера.

#  sysctl -q -n -w kernel.randomize\_va\_space=2

Добавьте строку ниже в /etc/sysctl.conf, если она еще не существует:

kernel.randomize\_va\_space = 2